TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

Deep Metric Learning-based Face Recognition Pipeline with Anti-spoofing on Raspberry-Pi Single-Board Computer

Rohit Anand[1], Abhishek Mann[2], Kundan Sharma [3]

[1,2,3]G.B.Pant Engineering College, New Delhi, India.

Article Info
Volume 82
Page Number: 4302 - 4308
Publication Issue:
January-February 2020

Article History
Article Received: *18 May 2019*
Revised: *14 July 2019*
Accepted: *22 December 2019*
Publication: *21 January 2020*

Abstract

Face recognition is an application of computer vision used for the identification of the persons present in an image / frame acquired through a camera sensor. Such a technology is already quite prevalent in multitude of cases such as security, biometric payments, augmented reality filters etc. In this paper, the concept, working and implementation of deep metric learning-based face recognition will be studied which is currently among state-of-the-art methods for face verification and recognition. Additionally, another module called anti-spoofing will be used to augment this face recognition pipeline to improve the security against photo and video based attacks that might be used by the malicious elements to spoof the face recognition system.

Keywords: *Face recognition, Face verification, Computer vision, Anti-spoofing, Deep metric learning.*

I. INTRODUCTION

Neural networks or multi layered perceptrons are formed by stacking layers of neurons sequentially and each neuron is densely connected with all the neurons that are present in the previous layer (for first hidden layer, the dense connections are present with the input layer). Each neuron in a layer is responsible for computing a linear function of its input and then applying a nonlinearity such as sigmoid/logistic [1], Rectified Linear Unit [2] to the linear function.

There are three types of layers in a neural network based on their position – input layer, hidden layer and output layer (shown in figure 1). The input layer refers to simply the input features stacked in a layer, output layer is the final layer of neural network and hidden layers are all those layers that are not either input or output layers. In supervised learning, the characteristics of the output layer change based on whether a regression or classification problem is being addressed. For a regression problem, the output layer will exist of a single neuron without any non-linearity applied after computation of linear function. Meanwhile for a classification problem, the output layer depends on whether if it is a binary classification problem or a multi-class classification problem.

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

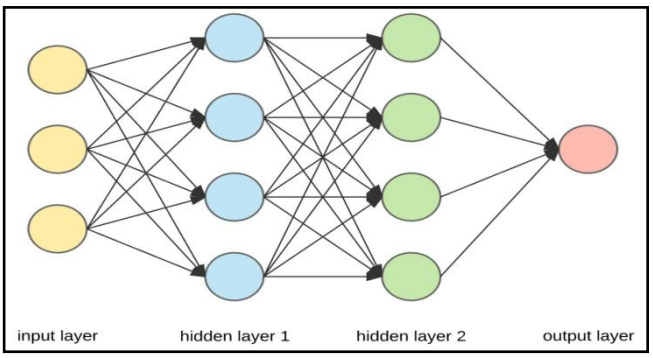

Fig 1. A neural network

In past, there have been many prevalent methods of face recognition such as eigenfaces. But recently, due to the meteoric rise of deep learning based methods, human level accuracy has reached on very challenging datasets such as Labeled Faces in the Wild (LFW). This paper focuses on the implementation of face recognition using a variant of ResNet-34 network architecture [3] using dlib and face_recognition library. Furthermore, an anti-spoofing approach based on treating real vs fake faces as a binary classification problem has been implemented. The anti-spoofing module used in this project is based on a convolutional neural network which is a special type of neural network that is considered state-of-the-art in the field of computer vision for the tasks such as image classification, object detection, semantic segmentation and instance segmentation. In this type of network, the convolution operation is performed between an image and a filter/kernel. The weights of these filters are treated as learnable parameters of the network. A widely known example of a revolutionary convolutional network is AlexNet [4]. There are two types of convolution operations – 'valid' and 'same'. For 'valid' convolution, no zero padding is done on the original image. While in 'same' convolution, zero padding is done such that the output has exactly the same spatial dimensions as input image.

Siamese network is a special type of neural network that contains two identical sub-networks with exactly same architecture, weights and hyperparameters. This network is used in face verification task for comparing the faces in two images to find out whether the faces in both images are of same person or not. When an image is given to one of the sub-networks in Siamese network, it calculates an output vector or embedding containing the encoding of the face in that image and this encoding is then compared with the encoding computed by giving other images containing a face to the other identical sub-network. This network is trained on triplet loss to minimize the Euclidean distance between the anchor-positive pairs and to increase the Euclidean distance between anchor-negative pairs [5].The Siamese network in our case is returning a 128-dimensional vector for every face present in the data set and during inference the network also generates 128-dimensional vector for each face present in the video frame.

Overfitting is a problematic condition where the network fits the training data but doesn't generalize to the test data or in other words it can be said that the network is suffering from high variance problem. In this condition, the network can give almost perfect accuracy over training data but performs poorly on examples the network has not seen before. This happens in a case where the network is overly complex for the training data it is trying to learn from. There are two ways to deal with this problem: Either getting more training data or using regularization. In this project, a regularization technique called dropout has been used. Dropout is a regularization technique used to decrease the overfitting problem in which the network creates overly complex decision boundary while learning from training data and fails to generalize to the data it has not seen during training [6]. Regularization methods basically work by restricting the values of weight parameters to small values so that model complexity is reduced and model is able to generalize well to the data that it has not seen during training. In dropout, the network drops units inside a layer based on the dropout probability. This decides about the probability of dropping a particular node

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

during every iteration of gradient descent. This helps in reducing the complexity of the network.

Batch normalization is generally used to deal with internal covariate shift arising while training neural networks [7]. Batch normalization allows the various layers of network to learn independently by decreasing the coupling between shallow and deeper layers. As the activation of earlier layers acts as input to deeper layers, what deeper layers learn is largely dependent on activations resulting from previous layers. Therefore, to decrease this coupling effect, the mean and variance of activations for different layers are restricted to specific values which are in-turn treated as learnable parameters of the network.

Face verification is considered a 1:1 problem where the algorithm has to differentiate whether the person in front of the camera are who they claim to be. Face recognition on other hand is a much harder problem due to the fact that it is a 1:K problem. In face recognition, the identity of the person has to be matched with 'K' different persons for whom the photos are available in the dataset. In hindsight, it might seem a simple task of using a convolutional neural network for face recognition but considering the scarcity of data that is available to us, it becomes impractical to train a convolutional neural network and furthermore, if a person has to be added or removed from the list of persons to be recognized by the algorithm, then the convolutional network must again be retrained. One possible way out of this adverse situation is by using state-of-the-art 'one shot learning' technique called deep metric learning for face recognition [8]. Another problem with modern face recognition system is their inability to withstand spoofing attacks.

Here, a possible anti-spoofing approach by treating spoofing as binary classification problem will be discussed followed by designing a convolutional neural network for differentiating between real vs. fake examples.

II. PROPOSED WORK

Here we propose the addition of an anti-spoofing method in the already available deep metric learning based face recognition pipeline from dlib. This anti-spoofing method is based on binary classification where our network will take an input frame and will clarify whether the face of the person in the image is real or is it spoofed like image of a person or video of that person.

The network architecture used for this binary classification task consists of convolutional, activation, batch normalization, pooling, dropout and fully connected layers. The exact architecture is discussed later on.

The dataset used for training this network is prepared by recording video of persons in different lighting conditions for generating positive examples and for negative examples the same video is recorded by another smartphone as this will be used for generating fake examples. Both of these videos are sampled at 4 frames and then the resulting frame is fed to Single Shot Detector (SSD) algorithm [9] for detection and cropping of the face in the frame. This gets our dataset ready.

The network above is constructed and trained using Keras deep learning framework which is using Tensorflow as backend. The network is trained for 65 epochs and the batch size is 8, the results are depicted in the results section. For optimization, we have used an advanced optimization algorithm called ADAM (Adaptive Moment Estimation).

III. IMPLEMENTATION

The implementation is broadly divided into three phases:

A. Face detection

B. Liveness detection/anti-spoofing

C. Face recognition

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

A. Face Detection

Face detection is a type of application classified under "computer vision" technology. It is the process in which algorithms are developed and trained to properly locate faces or objects (in object detection, a related system) in images. These can be in real time from a video camera or from photographs. An example where this technology is used are in airport security systems. In order to recognize a face, the camera software must first detect it and identify the features before making an identification. Likewise, when Facebook makes tagging suggestions to identify people in photos, it must first locate the face. On social media apps like Snapchat, face detection is required to augment reality which allows users to virtually wear dog face masks using fancy filters. Another use of face detection is in smartphone face ID security.

Face detection uses classifiers which are algorithms that detect what is either a face(1) or not a face(0) in an image. Classifiers have been trained to detect faces using thousands to millions of images in order to get more accuracy. OpenCV uses two types of classifiers - LBP (Local Binary Pattern) and Haar Cascades. The latter classifier will be used.

Haar Cascades is based on the Haar Wavelet technique to analyze pixels in the image into squares by function. This uses machine learning techniques to get a high degree of accuracy from what is called training data. This uses "integral image" concepts to compute the features detected. Haar Cascades use the Adaboost learning algorithm that selects a small number of important features from a large set to give an efficient result of classifiers [10].

B. Liveness Detection

In liveness detection or anti-spoofing, the network is determining whether the person in the video frame is an actual person or a malicious element is trying to spoof the face recognition pipeline [11]. Architecture of anti-spoofing network is shown in figure 2.

For this, we have considered spoofing as a binary classification problem in which the convolution neural network will output two values using a softmax output unit which will be then used to determine whether the person is real or fake.

```
model.add(Conv2D(16, (3, 3), padding="same", input_shape=inp_shape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chan_dim))
model.add(Conv2D(16, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chan_dim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(32, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chan_dim))
model.add(Conv2D(32, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chan_dim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(64))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))

model.add(Dense(classes))
model.add(Activation("softmax"))
```

Fig 2. Anti–spoofing Network Architecture

The network used for liveness detection is a four layered architecture in which each layer contains a convolutional layer followed by activation layer and then a batch normalization layer, pooling and dropout layers have also been used. For creating the network in python environment, Keras deep learning framework has been used. Keras is a deep learning framework that uses Tensorflow as backend. Tensorflow is a differentiable programming library that is used for generating and executing computation graphs. It is mostly used for defining and training machine learning models

C. Face Recognition

In face recognition part, the facial embedding of detected face is generated (128-d vector). The Euclidean distance is computed pair wise between the embedding in concern and the embeddings that are generated from face images in the dataset. If the Euclidean distance between any such case is less than or equal to 0.5, then the person is recognized as that person for whose face this condition holds true. In this project, face recognition has been implemented using dlib and face_recognition libraries. The face_recognition is a wrapper around dlib and allows for more effective usage of dlib in

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

construction of face recognition pipelines. For making the face recognition pipeline more computationally efficient, the face embeddings for images present in dataset are pre-computed using any one of the sub-networks of Siamese network and then the resulting encodings/embeddings are serialized and stored in a '.pickle' file. So during inference, instead of computing encodings for the images present in dataset in real-time, this '.pickle' file is used that contains pre-computed encodings. As only one sub-network of Siamese network will be used, so the face recognition pipeline will be far more computationally efficient.

IV. RESULTS

The figure 3 (Learning Curve 1) depicts the change in categorical cross entropy loss as the number of epochs increases. As the loss decreases for the more number of epochs, it is fair to state that Adaptive Moment Estimation Optimizer is successful in finding a reasonable minimum to the optimization problem.

It may be seen from the figure 4 (Learning Curve 2) that the overall trend depicts the improvement in accuracy of the model as the number of epochs keep increasing and hence it is fair to judge that the model is learning a reasonable mapping. For epoch 65, 99.25% accuracy is observed on the holdout cross-validation set.

As it can be seen that cross-validation loss is considerably less than training loss, this happens because a regularization technique called dropout is being used. Dropout operates such that random nodes in a layer are being eliminated at every iteration of gradient descent only for training but at test time dropout is turned off and then the result is being computed. The primary criteria for judging the performance of this network is accuracy and loss on cross-validation set, as cross-validation set consists of examples that the network has not been trained on.

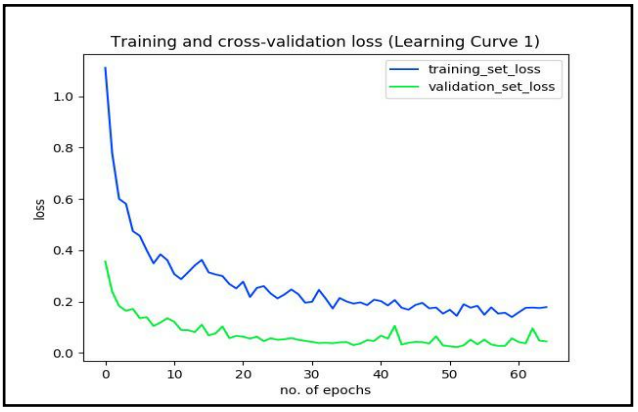

Fig 3. Learning Curve 1

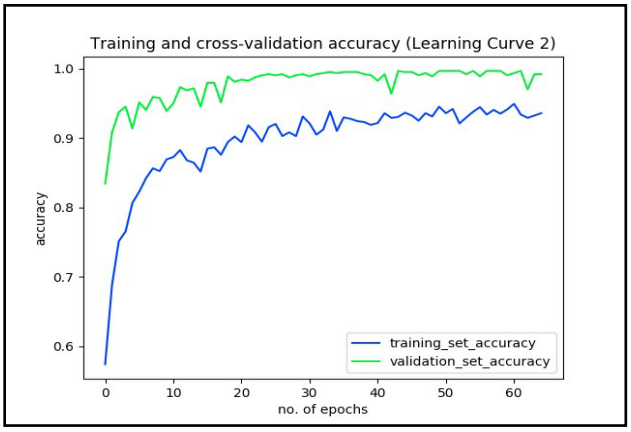

Fig 4. Learning Curve 2

From figure 5, it may be noticed how the anti-spoofing model is able to distinguish that the person in frame is real with the probability of 0.9962. Furthermore, the face of the person has been recognized successfully as the Euclidean distance between encoding of face present in the frame and pre-computed encoding of the individual concerned is less than the tolerance value i.e. 0.5. From figure 6, it may be observed that the face recognition pipeline handles the case accurately even when two distinct faces are present in the frame. With faces being recognized correctly and also distinguished as 'real' and 'fake' (from left to right). In figure 7, it is seen that the person is recognized successfully as "abhishek_mann" with the probability of the face being 'fake' as 0.9069. From figure 8, it is seen that the person is recognized successfully as "kundan_sharma" with the probability of the face being 'fake' as 0.9267.

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

The above images show the results obtained from the face recognition pipeline. Notice how the anti-spoofing module is able to distinguish between real and fake faces. Furthermore the faces in the frame are being recognized fairly well despite changes in lighting conditions.

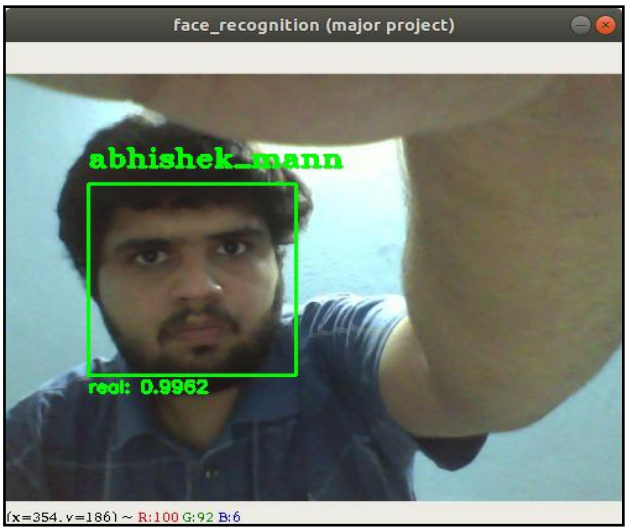

Fig 5. Output 1

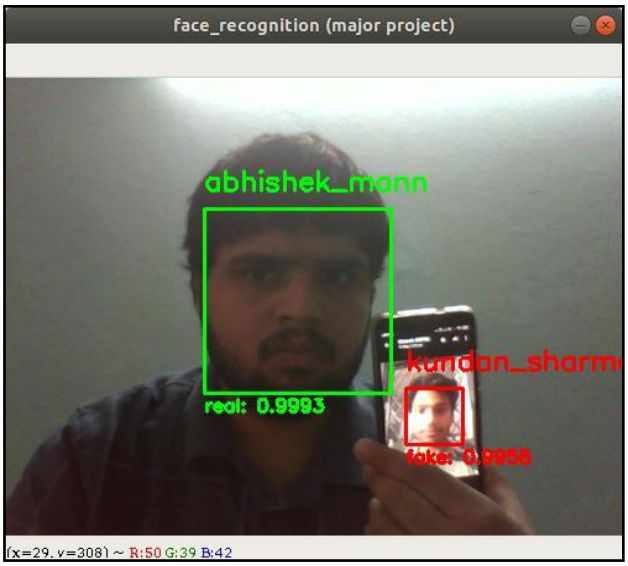

Fig 6. Output 2

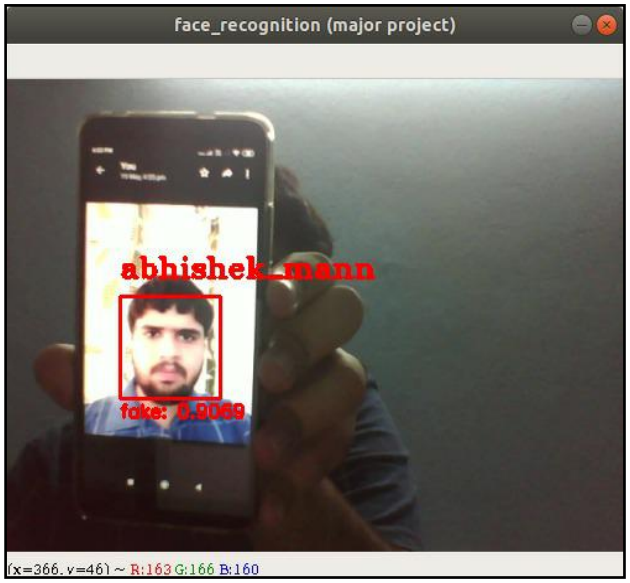

Fig 7. Output 3

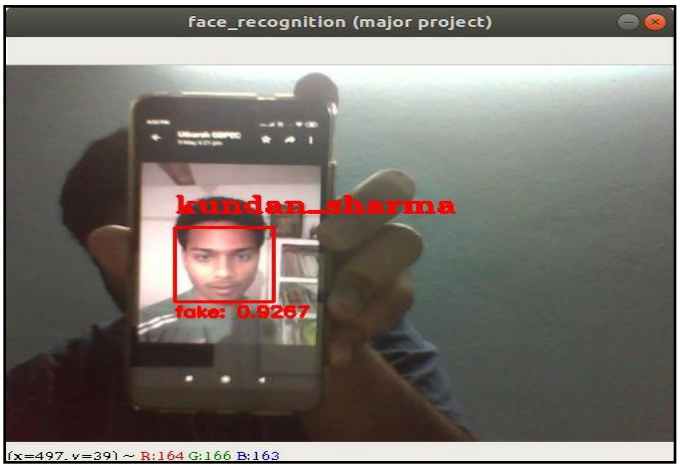

Fig 8. Output 4

V. MAJOR APPLICATIONS

1. Law enforcement and surveillance – In Baidu (Chinese search engine giant), instead of handing RFID cards to the employees, there is a turnstile based entrance system in which face recognition is used. The face recognition pipeline used there is robust to spoofing attacks and only a genuine employee is allowed entrance to restricted areas. Meanwhile very recently in San Francisco the use of face recognition technology has been banned to avoid dystopian circumstances.

2. Smart payments – Confirming payments via face recognition is a hot topic at present. Using face recognition technology for authenticating payments will allow users to ditch ATM cards etc.

January - February 2020
ISSN: 0193 - 4120 Page No. 4302 - 4308

3. AR (Augmented Reality) apps – Popular applications such as Snapchat and Instagram use face detection, recognition and facial landmarks detection for applying fancy filters.

4. AI powered IoT devices – With the advent of increasingly powerful SBCs such as NvidiaJetson Nano and Google Coral, it is possible to deploy AI powered applications at the edge (on premises) for single board computers and therefore building device smarter than ever before.

5. Autonomous vehicles – In autonomous vehicles being developed by companies such as Tesla, major focus is on driver activity recognition. So driver can be notified when they should take control of the vehicle depending on a situation. To make this as non-intrusive, the possible pure computer vision based methods are used. Face recognition plays major role in such a sophisticated system.

VI. CONCLUSION

The face recognition pipeline thus created is able to recognize the faces of different people in distinct lighting conditions and also differentiates whether the face in front of the camera is real or the face recognition pipeline being attacked by photo or video based spoofing attack.

REFERENCES

[1] [1]J. Han and C. Moraga, From Natural to Artificial Neural Computation. Springer, 1995, pp. 195-201.

[2] [2]K. Hara, D. Sato and H. Shouno, "Analysis of function of rectified linear unit used in deep learning", in International Joint Conference on Neural Networks, Killarney, Ireland, 2015.

[3] [3]K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", in International Conference on Machine Learning, New York City, 2016.

[4] [4]A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, Lake Tahoe, Nevada, 2012, pp. 1097-1105.

[5] [5]F. Schroff, J. Philbin and D. Kalenichenko, “FaceNet: A Unified Embedding for Face Recognition and Clustering” in 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, Massachusetts, 2015.

[6] [6]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.

[7] [7]S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", in Proceedings of 32nd International Conference on Machine Learning, Lille, France, 2012, pp. 448-456.

[8] [8]Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", in 27th IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, 2014.

[9] [9]W. Liu et al., Computer Vision - ECCV 2016. Springer, 2016, pp. 21-37.

[10] [10]P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", in Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001.

[11] [11]S. Chakraborty and D. Das, "AN OVERVIEW OF FACE LIVENESS DETECTION", International Journal on Information Theory, vol. 3, no. 2, pp. 11-24, 2014.